

PRVPATENT- OCH REGISTRERINGSVERKET
Patentavdelningen

24 -08- 2004

**Intyg
Certificate**

REC'D 02 SEP 2004

WIPO

PCT

Härmed intygas att bifogade kopior överensstämmer med de handlingar som ursprungligen ingivits till Patent- och registreringsverket i nedannämnda ansökan.

This is to certify that the annexed is a true copy of the documents as originally filed with the Patent- and Registration Office in connection with the following patent application.



(71) Sökande Anette Bergström, Solna SE
Applicant (s)

(21) Patentansökningsnummer 0302227-4
Patent application number

(86) Ingivningsdatum 2003-08-15
Date of filing

Stockholm, 2004-08-20

För Patent- och registreringsverket
For the Patent- and Registration Office

Marita Öun

Avgift
Fee

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

Metod och arrangemang för att spåra utförda operationer vid användning av dataprogram

Uppfinningens område

- 5 Föreliggande uppfinning avser en metod och ett arrangemang för att spåra operationer eller aktiviteter som utförts vid användning av ett dataprogram. Speciellt avses en lösning för att kunna spåra eventuella felaktiga inmatningar och/eller kommandon gjorda av en användare.

10

Teknikens bakgrund

- Vid arbete med olika dataprogram och datorapplikationer är det inte ovanligt att ett program plötsligt låser sig eller att detta inte arbetar på ett sätt som önskas eller förväntas.
- 15 Detta kan bero på att en användare, mer eller mindre omedvetet, har utfört någon felaktig eller olämplig operation i programmet, eller att det finns något fel i detta, en sk. "bugg". Sådana operationer utförs vid aktivering av olika funktioner, knapptryckningar, musklickningar, etc, som när de
- 20 utförs enskilt eller i kombination kan medföra låsning eller felaktig funktion hos det eller de program som används. I denna beskrivning kommer uttrycket "operation" att användas för att beteckna alla tänkbara åtgärder som en användare kan utföra i ett dataprogram och som på något sätt
- 25 påverkar/aktiverar detta. En utförd operation är således ett resultat av en eller flera inmatningar, såsom tangenttryckningar, musklickningar eller tal vid röststyrda program.

- Vid exempelvis användning av dataprogram för
- 30 ordbehandling, kalkylblad eller databas kan ibland helt oväntade situationer uppstå, såsom att data förloras, utseendet på skärmen ändras, eller att programmet låser sig så att det inte svarar på några inmatningar. För att kunna vidta nödvändiga åtgärder som rättar till programmets "beteende" är

det ofta till stor hjälp om tidigare och hittills utförda operationer är kända och kan analyseras. Om således en tidigare gjord felinmatning kan spåras är det lättare att förstå vad som har hänt och varför den nuvarande situationen har uppstått.

Vid dataundervisning för användning av speciella dataprogram är det brukligt att en lärare hjälper och instruerar elever som tränar sig på att använda programmet. Det är då nästan oundvikligt att situationer som nämns ovan

förr eller senare uppstår. Läraren frågar ofta eleven vad han/hon har gjort i programmet innan exempelvis ett fel uppstod. Ofta minns då inte eleven alla operationer som utförts, eller också har vissa operationer utförts helt oavsiktligt eller omedvetet. Felet kan t.ex. bero på någon inmatning som gjorts tämligen långt bak i tiden, eller på att eleven helt enkelt råkat trycka ned en tangent utan att märka det. Läraren måste då ofta ägna lång tid åt att ta reda på vad som har hänt och vad som behövs för att rätta till situationen. Onödiga diskussioner kan även uppstå om vad eleven har gjort och inte gjort. Dessutom är det önskvärt att eleven får veta vilket eller vilka misstag som begåtts för att kunna undvika dessa i framtiden och lära sig hur programmet fungerar.

De ovan nämnda problemen och behoven kan förekomma även i många andra situationer, såsom på företag eller organisationer där datoranvändare kan vända sig till en person eller avdelning som tillhandahåller information och hjälp vid frågor om dataprogram, sk. "datasupport", eller för enskilda användare som vill kunna göra snabba felsökningar eller liknande. Det kan även vara användbart att kunna spåra tidigare utförda operationer vid test och utvärdering av ett nytt eller modifierat dataprogram, för att hitta eventuella brister och felprogrammeringar.

US 2001/0003172 A1 beskriver ett system där en användares olika datoraktiviteter registreras för att skapa en loggfil som är avsedd att användas för att bokföra användarens arbete inom olika projekt mm. Detta US-dokument beskriver dock inte spårning av enskilda operationer utförda vid användning av ett specifikt dataprogram, vilket ej heller efterstävas eller behandlas i lösningen enligt ovan nämnda skrift.

Det finns således allmänt ett behov att på ett enkelt sätt kunna se och identifiera alla enskilda operationer som utförts vid användning av specifika dataprogram, för att exempelvis upptäcka eventuella felaktiga inmatningar och/eller kommandon gjorda av en användare.

Redogörelse för uppfinningen

Det är ett syfte med föreliggande uppfinning att erhålla en lösning som medger spårning av utförda enskilda operationer vid användning av specifika dataprogram.

Detta syfte och andra uppnås genom en metod och ett arrangemang för att spåra enskilda operationer som utförts i ett specifikt dataprogram, där inmatningar detekteras som utförs av en användare. Varje detekterad inmatning jämförs med en uppsättning på förhand definierade operationer, varvid en mot varje inmatning svarande operation registreras som en utförd aktivitet om inmatningen matchar någon fördefinierad operation vid jämförelsen. Proceduren repeteras för ett antal inmatningar utförda under en inspelningssession, varvid en aktivitetslista med utförda operationer skapas. Denna lista kan sedan tas fram efter inspelningssessionen för att spåra enskilda operationer.

Kortfattad beskrivning av ritningarna

Föreliggande uppfinning beskrivs närmare nedan med hjälp av föredragna exempel på utföringsformer och med hänvisning till de bilagda ritningarna:

Fig. 1 är ett schematiskt blockschema för en systemarkitektur av ett administrativt verktyg som kan användas för föreliggande uppfinning,

Fig. 2 är ett schematiskt blockschema för en systemarkitektur av ett spårningsprogram som kan användas för föreliggande uppfinning, och

Fig. 3 är ett flödesschema över ett arbetssätt för bestämning av tidigare utförda operationer i ett specifikt dataprogram enligt föreliggande uppfinning.

10

Detaljerad beskrivning av föredragna utföringsformer

Föreliggande uppfinning avser att åstadkomma snabb och effektiv spårning av tidigare utförda operationer i ett dataprogram, för att exempelvis underlätta upptäckande av eventuellt felaktiga inmatningar eller kommandon som en användare utfört, eller av eventuella felprogrammeringar eller brister i programmet.

I korthet innebär uppfinningen att de operationer som en användare utför registreras och lagras, dvs. "spelas in", under den tid som ett bevakat dataprogram används. För detta ändamål används ett speciellt utformat program, vilket för enkelhets skull fortsättningsvis kommer att kallas "spårningsprogrammet", som körs parallellt med dataprogrammet och bevakar detta med avseende på användarens aktiviteter.

Vid inspelning av användarens aktiviteter identifieras utförda operationer genom att varje inmatning eller kombination av inmatningar detekteras och jämförs med en uppsättning på förhand definierade operationer som på något sätt kan påverka programmet. Dessa fördefinierade operationer finns således lagrade tillgängligt för spårningsprogrammet. Sådana fördefinierade operationer kan vara dels allmängiltiga oberoende av vilket program som används, såsom nedtryckning av fysiska tangenter på ett tangentbord, eller specifika för varje program, såsom val av programmets tillgängliga

30

funktioner exempelvis genom klickning på ikoner, menyer, knappar, etc. som visas på skärmen. I många program kan även vissa kommandon utföras genom nedtryckning av en eller flera tangenter, såsom exempelvis Ctrl+S för att spara dokument, eller funktionstangenter F1-F12.

Spårningsprogrammet innefattar således en funktion som detekterar användarens inmatningar och jämför dessa med de fördefinierade operationerna. Om en detekterad inmatning matchar en sådan fördefinierad operation registreras denna som en utförd aktivitet. Spårningsprogrammet är således utformat att successivt registrera på så sätt identifierade operationer i samband med att dessa utförs, och spara dessa i en aktivitetslista. Denna lista kan även företrädesvis innehålla uppgift om tidpunkter då operationerna utfördes.

Vid ett senare tillfälle, när inspelningssessionen har avslutats, kan aktivitetslistan med sparade operationer tas fram och genomgå för att analysera händelseförloppet och spåra enskilda operationer, exempelvis eventuell felinmatning. Detta kan vara aktuellt när programmet har låst sig eller inte fungerar som det ska, eller om en användares färdighet skall utvärderas efter t.ex. en utförd test. Inom ramen för uppfinningen kan aktivitetslistan utformas på olika sätt. De registrerade operationerna kan t.ex. sorteras kronologiskt i listan eller på annat sätt. Operationerna kan även klassificeras och sorteras i olika grupper exempelvis beroende på typ av operation, vilket kan underlätta sökning och analys i vissa situationer.

Operationerna i listan är företrädesvis benämnda med förutbestämda lämpliga beteckningar i klartext för att lätt kunna identifieras och förstås, till skillnad mot de beteckningar som används av datorns operativsystem. Dessa operationsbeteckningar bestäms lämpligen vid fördefinieringen av operationerna enligt ovan. En sådan operationsbeteckning kan valfritt bestå av ett kortfattat namn eller en längre mer

informativ beskrivning, allt efter behov. Vidare kan olika filterfunktioner användas, så att endast vissa typer av operationer registreras medan andra ignoreras. Exempelvis kan teckeninmatningar filtreras bort i det fall dessa inte

5 påverkar programmet i en viss situation.

Spårningsprogrammet kan vara utformat för att kunna bevaka flera olika dataprogram, genom att innehålla en fil för varje program med fördefinierade operationer och eventuella filterfunktioner. Det är även möjligt att installera en ny
10 sådan fil när ett nytt program skall läggas till för bevakning, eller då ett bevakat program har modifierats.

Med hänvisning till Fig. 1 och 2 kommer ett exempel på en lämplig systemarkitektur att beskrivas som kan användas för att implementera föreliggande uppfinning. Systemarkitekturen
15 kan i princip delas upp i ett administrativt verktyg och ovan nämnda spårningsprogram, där det administrativa verktyget används huvudsakligen för att definiera och lagra de operationer som kan utföras vid användning av ett bevakat program.

20 Fig. 1 visar ett logiskt blockschema för ett administrativt verktyg 100 som kan användas av exempelvis en systemleverantör för att bygga upp en operationsdatabas 102 med fördefinierade operationer för ett specifikt dataprogram P som senare skall bevakas. Allmänt gäller att när olika inmatningar utförs i
25 programmet P genereras på känt sätt olika meddelanden av programmet som skickas till andra program eller applikationer i datorn för att resultera i olika händelser, såsom nytt utseende på skärmen. Varje inmatning som påverkar programmet motsvaras således av ett specifikt meddelande. Sådana
30 genererade meddelanden kan i praktiken avlyssnas, exempelvis med hjälp av en känd funktion kallad "SetWindowsHookEx()" som ingår i det välkända operativsystemet Windows, vilken kan utnyttjas i detta sammanhang.

Det administrativa verktyget 100 innefattar en loggningsfunktion 104 som är utformad att avlyssna och fånga upp sådana meddelanden som genereras av programmet P som svar på inmatningar, och tolka dessa. I praktiken utförs

- 5 systematiskt en grunduppsättning av operationer i programmet P som en användare kan tänkas utföra, varvid loggningsfunktionen 104 loggar meddelanden som fångas upp i samband med varje operation. Dessa sparas då som olika mönster i operationsdatabasen 102, där varje fördefinierad operation
- 10 svarar mot ett eller flera specifika genererade meddelanden.

- Det administrativa verktyget 100 innefattar vidare en benämningsfunktion 106 genom vilken de sparade operationerna kan namnges. Benämningsfunktionen 106 styrs genom ett användargränssnitt, ej visat, i vilket en systemadministratör
- 15 eller motsvarande kan manuellt definiera en valfri beteckning, såsom ett namn eller en beskrivning, för varje utförd operation. Denna beteckning är avsedd att lätt kunna identifieras och förstås för en framtida användare av spårningsprogrammet. De registrerade operationerna sparas
- 20 slutligen tillsammans med motsvarande beteckning i operationsdatabasen 102. Alternativt kan operationerna sparas som en enkel kod i operationsdatabasen 102 medan de definierade motsvarande beteckningarna sparas i en separat regeldatabas som anger hur informationen skall presenteras för
- 25 slutanvändaren, vilken kommer att beskrivas nedan.

Den ovan beskrivna proceduren utförs i princip för alla tänkbara och/eller intressanta operationer för det specifika dataprogrammet P för att bygga upp operationsdatabasen 102, vilket således kan ses som ett slags kalibreringsförfarande.

30 När operationsdatabasen 102 är komplett kan programmet bevakas genom inspelning av identifierade operationer med hjälp av spårningsprogrammet, vilket beskrivs närmare nedan.

Fig. 2 visar ett logiskt blockschema för en lämplig arkitektur för spårningsprogrammet 200 som är avsett att köras

i samband med att dataprogrammet P skall användas och bevakas. Spårningsprogrammet 200 innefattar en serie logiska funktioner som utförs i tur och ordning enligt nedan. En loggningsfunktion 202 är utformad att fånga upp olika meddelanden som genereras i programmet P när en användare utför olika inmatningar, på motsvarande sätt som beskrivits i samband med Fig. 1. Denna loggningsfunktion 202 kan således vara av samma typ som den loggningsfunktion 104 som använts i det administrativa verktyget 100, exempelvis ovan nämnda kända funktion

10 "SetWindowsHookEx()".

Loggade inmatningar behandlas sedan av en analysfunktion 204 där onödig information, eller "brus", kan sorteras bort. Detta kan vara meddelanden som genereras i samband med inmatningar som inte bedöms vara intressanta, t.ex. en så kallad "fokusfunktion" som innebär att fokus kan flyttas mellan olika menyer eller knappar som markeras på skärmen genom nedtryckning av piltangenter eller motsvarande.

Därefter matas den mottagna informationen till en matchningsfunktion 206 som jämför utförda inmatningar med de fördefinierade operationerna som lagrats i operationsdatabasen 102 enligt Fig. 1. Om en operation då identifieras som matchar utförd inmatning, kan denna operation även behandlas av en presentationsfunktion 208 som är utformad att tillämpa fördefinierade presentationsregler på de identifierade operationerna. Dessa presentationsregler är lagrade i en regeldatabas 210. Sådana regler kan vara språklig kontroll samt tillämpning av speciella önskemål från slutanvändaren om hur informationen skall presenteras, etc.

Presentationsreglerna kan också innefatta en eller flera filterfunktioner så att endast vissa typer av operationer registreras medan andra ignoreras. Operationernas beteckningar kan även vara lagrade i regeldatabasen 210, såsom nämnts ovan.

Presentationsfunktionen 208 är således avsedd att spara informationen i ett format som är begripligt och motsvarar

slutanvändarens önskemål. Inom ramen för denna uppfinning kan flera olika användare definiera sina egna profiler vilka sparas i denna regeldatabas 210. Beroende på vilken användare som är inloggad för en viss inspelningssession, tillämpas de
5 regler som motsvarar denna användare.

Slutligen sparas den bearbetade informationen om utförda operationer i en aktivitetsdatabas 212 som i efterhand kan konsulteras för att spåra enskilda operationer, såsom beskrivits ovan. Spårningsprogrammet 200 är även försett med
10 ett lämpligt användargränssnitt, ej visat, som kan användas för att gå igenom de operationer som tidigare utförts under en inspelningssession som nu avslutats. Detta spårningsgränssnitt kan lämpligen visa en förteckning av vilka dataprogram som kan bevakas med spårningsprogrammet, funktionsknappar för
15 hantering av inspelning, såsom start, stopp och paus, status för en eller flera pågående eller avslutade inspelningssessioner, detaljerade aktivitetslistor över utförda operationer för varje session, dialogrutor för inloggning och definition av användarprofiler, för att nämna
20 några exempel.

Figur 3 visar ett förenklat flödesschema över den procedur som spårningsprogrammet genomgår för att registrera utförda operationer efter att inspelningsfunktionen aktiverats. I ett första steg 300 detekteras en av en användare utförd
25 inmatning. Inom ramen för denna uppfinning kan en sådan inmatning exempelvis bestå av en knapptryckning, en musklickning eller ett talkommando, utförda enskilt eller i kombination. Denna inmatning jämförs sedan, i ett steg 302, med de fördefinierade operationerna som finns lagrade i
30 operationsdatabasen 102. I nästa steg 304 bestäms om inmatningen matchar någon av de fördefinierade operationerna. Om så inte är fallet återgår proceduren till steg 300 för att undersöka en ny inmatning. Om däremot en matchande operation identifieras i steg 304 tas en motsvarande beteckning fram för

denna operation, i ett nästa steg 306, antingen från operationsdatabasen 102 eller från regeldatabasen 210 beroende på var den har lagrats under den ovan beskrivna kalibreringsproceduren. Slutligen lagras denna operationsbeteckning i aktivitetsdatabasen 212, i ett steg 308, eventuellt efter en anpassning enligt uppgifter i regeldatabasen 210 såsom beskrivits ovan.

5 Föreliggande uppfinning gör det möjligt att på ett enkelt och smidigt sätt kunna analysera ett händelseförlopp i detalj vid användning av ett specifikt bevakat dataprogram. Således registreras alla enskilda operationer vilka identifierats som utförda under en inspelningssession, baserat på användarens inmatningar, och som på förhand definierats som relevanta för det bevakade programmet. Denna lösning kan användas exempelvis 15 för att upptäcka eventuella felaktiga inmatningar och/eller kommandon, eller för att utvärdera ett nytt eller modifierat program.

Uppfinningen är inte begränsad till de ovan beskrivna utföringsformerna, utan definieras allmänt av de efterföljande 20 patentkraven.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

Patentkrav

1. En metod att spåra enskilda operationer som utförts i ett specifikt dataprogram, **kännetecknad av** följande steg att:

- 5 A) detektera minst en inmatning utförd av en användare,
B) jämföra detekterad inmatning med en uppsättning på förhand definierade operationer,
C) registrera en mot inmatningen svarande operation som en utförd aktivitet, om inmatningen matchar någon
10 fördefinierad operation vid jämförelsen utförd i steg B),
och
D) repetera stegen A) - C) för ett antal inmatningar utförda under en inspelningssession, varvid en aktivitetslista med utförda operationer skapas, så att
15 nämnda lista kan tas fram efter inspelningssessionen för att spåra enskilda operationer.

2. En metod enligt krav 1, **kännetecknad av** att en inmatning detekteras i steg A) genom avlyssning av ett motsvarande
20 meddelande som genereras av dataprogrammet som svar på inmatningen.

3. En metod enligt krav 1, **kännetecknad av** att nämnda uppsättning fördefinierade operationer har lagrats i en
25 operationsdatabas.

4. En metod enligt krav 3, **kännetecknad av** att nämnda uppsättning innefattar operationer som är allmängiltiga oberoende av vilket program som används eller specifika för
30 programmet.

5. En metod enligt krav 3 eller 4, **kännetecknad av** att nämnda uppsättning fördefinierade operationer har skapats genom att utföra en grunduppsättning av operationer som en

användare kan tänkas utföra i programmet, för att bygga upp nämnda operationsdatabas.

- 5 6. En metod enligt något av kraven 1 - 5, **kännetecknad av** att registrering av en utförd operation i steget C) innefattar att ta fram en fördefinierad beteckning för operationen, vilken beteckning lagras i nämnda aktivitetslista.
- 10 7. En metod enligt något av kraven 1 - 6, **kännetecknad av** att fördefinierade presentationsregler tillämpas på utförda operationer innan dessa lagras i nämnda aktivitetslista.
- 15 8. En metod enligt krav 7, **kännetecknad av** att nämnda presentationsregler har lagrats i en regeldatabas.
- 20 9. En metod enligt krav 7 eller 8, **kännetecknad av** att nämnda presentationsregler innefattar en filterfunktion så att endast vissa typer av operationer registreras medan andra ignoreras.
- 25 10. Ett arrangemang för att spåra enskilda operationer som utförts i ett specifikt dataprogram, **kännetecknat av**:
- medel för att skapa en uppsättning på förhand definierade operationer,
- medel för att detektera inmatningar utförda av en användare och jämföra dessa med nämnda uppsättning fördefinierade operationer,
- medel för att registrera operationer som utförda aktiviteter under en inspelningssession för de inmatningar som matchar någon fördefinierad operation, för att skapa en aktivitetslista med utförda operationer, så att nämnda lista kan tas fram efter inspelningssessionen för att spåra enskilda operationer.
- 30
35

Sammandrag

En metod och arrangemang för att spåra enskilda operationer som utförts i ett specifikt dataprogram (P), där inmatningar detekteras som utförs av en användare. Varje

5 detekterad inmatning jämförs med en uppsättning på förhand definierade operationer, vilka har lagrats i en operationsdatabas (102). Om en detekterad inmatning matchar någon fördefinierad operation vid jämförelsen registreras denna operation som en utförd aktivitet i en

10 aktivitetsdatabas (212). Proceduren repeteras för ett antal inmatningar utförda under en inspelningssession, varvid en aktivitetslista med utförda operationer skapas i aktivitetsdatabasen. Denna lista kan sedan tas fram efter inspelningssessionen för att spåra enskilda operationer.

15 Därmed kan ett händelseförlopp analyseras i detalj efter användning av dataprogrammet, exempelvis i syfte att upptäcka felaktiga operationer.

(Fig. 2)

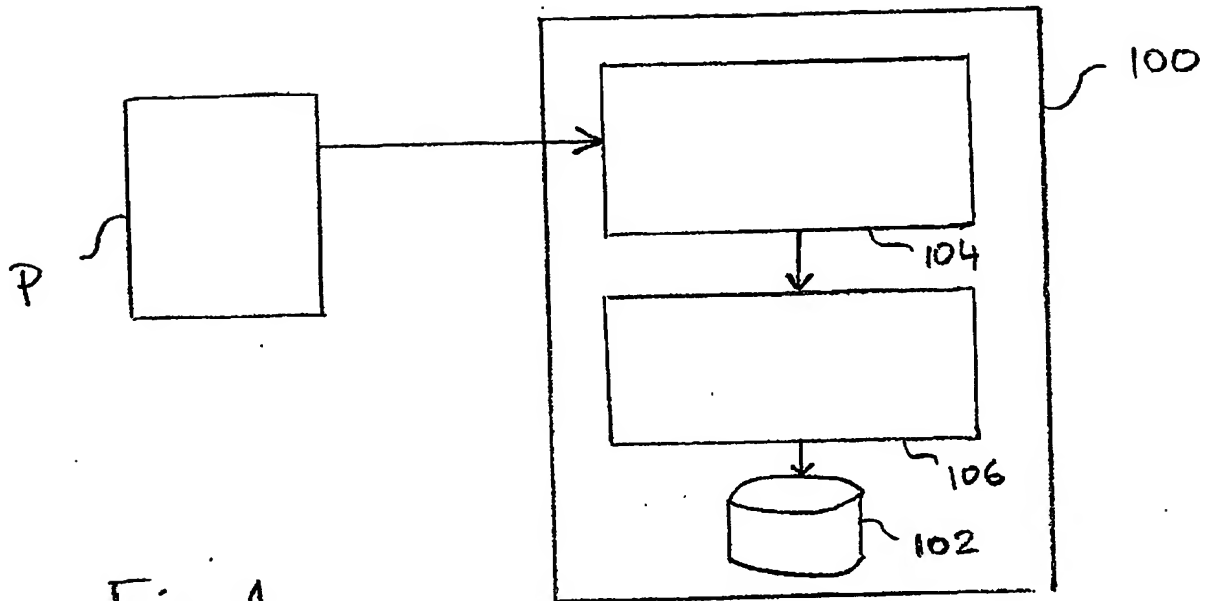


Fig. 1

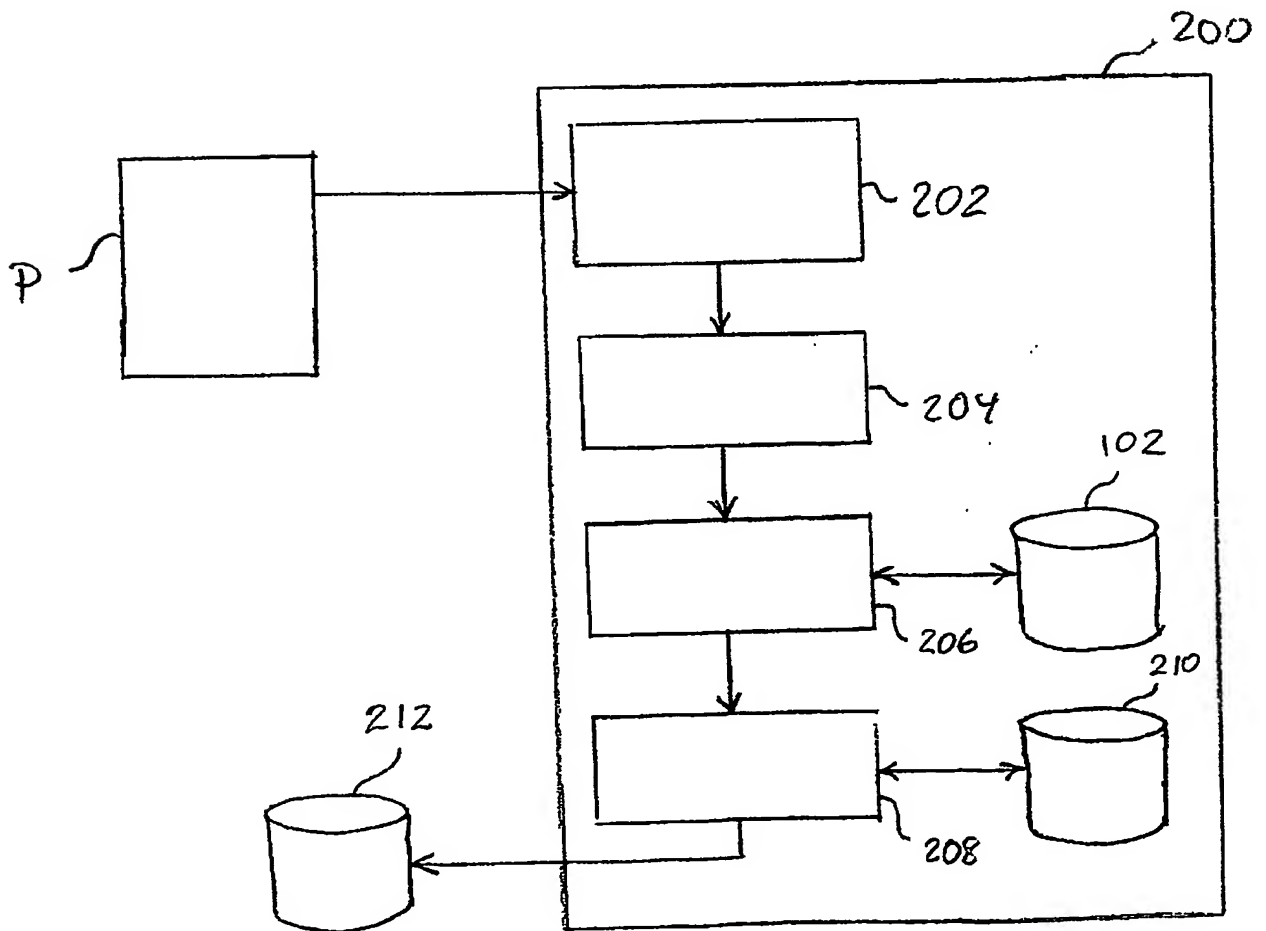


Fig. 2

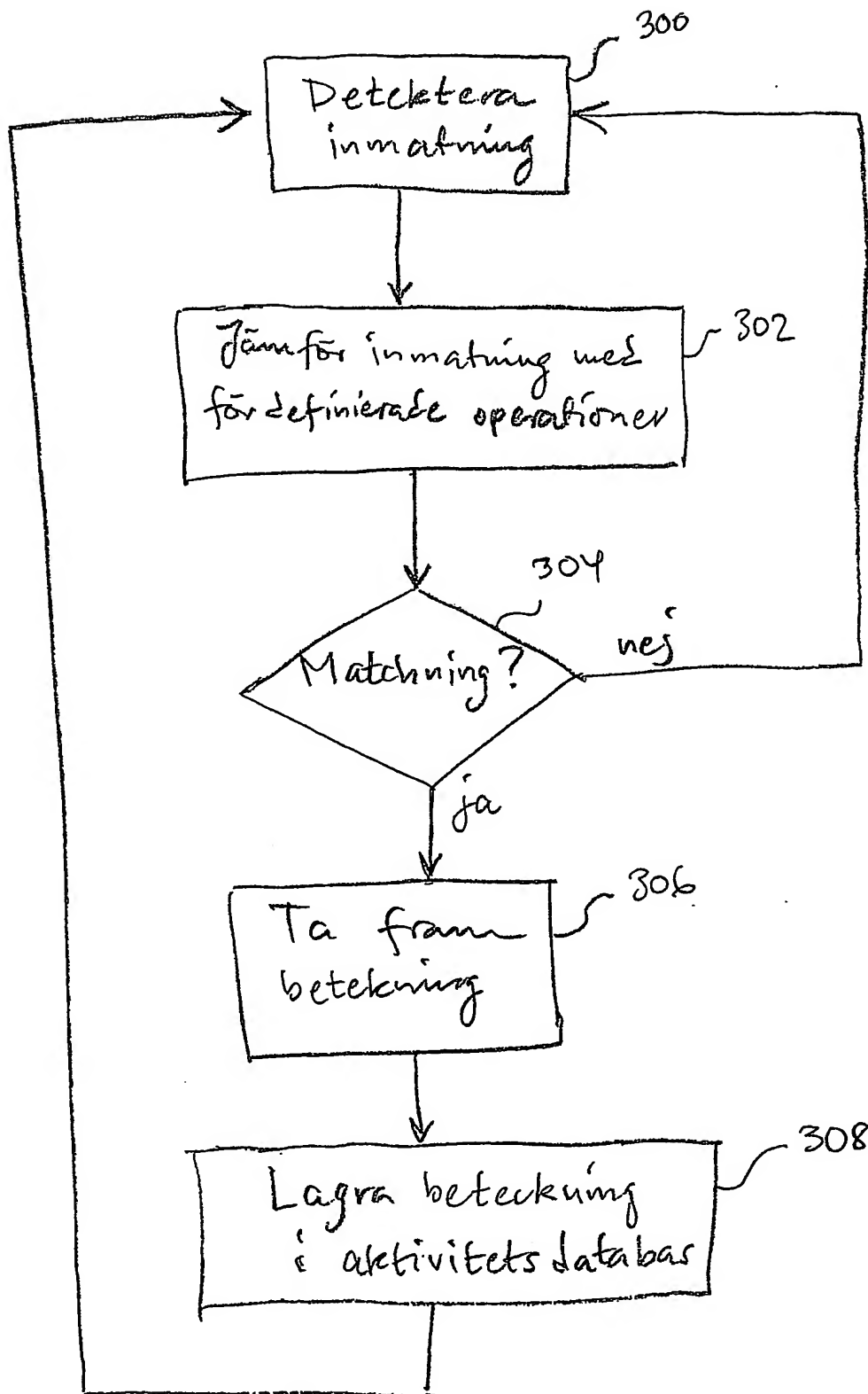


Fig. 3